

AMENDMENTS TO THE CLAIMS

For the convenience of the Examiner, all claims have been presented whether or not an amendment has been made. The claims have been amended as follows:

1. **(Amended)** A source code line counting system, comprising:
a computer readable storage medium; and
a software counting tool stored on the computer readable storage medium and operable to
 parse a first file containing computer source code to create a token stream in response to one of a plurality of sets of configuration data,
 wherein the computer source code in the file was written in one of a plurality of computer languages that may be processed by the software counting tool,
 wherein each set of configuration data comprises keywords for one or more of the plurality of computer languages,
 create a list of statements in response to the token stream, **at least some of the statements comprising a combination of two or more tokens**, and
 generate a count value in response to the list of statements.
2. **(Original)** The system of Claim 1, wherein the plurality of sets of configuration data includes a different set for each of the following computer programming languages: C++ and Cobol.
3. **(Original)** The system of Claim 1, wherein the plurality of sets of configuration data includes a different set for at least two of the following computer programming languages: C, C++, Java, Fortran, Cobol, and Basic.
4. **(Original)** The system of Claim 1, wherein each token in the token stream comprises a string of one or more characters and wherein each token in the token stream is associated with a token type value.

5. **(Original)** The system of Claim 4, wherein the token type value comprises a data value indicating that the token is of a type selected from the group comprising: an operator, a comment, a constant, a keyword, and an identifier.

6. **(Original)** The system of Claim 4, wherein the possible token type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

7. **(Original)** The system of Claim 1, wherein portions of the token stream are ignored in generating the statement list because such portions are associated with a portion of the source code in the first file that is written in a language different from the one of the plurality of computer languages.

8. **(Original)** The system of Claim 1, wherein each statement in the list of statements is further associated with a statement type value.

9. **(Original)** The system of Claim 8, wherein the possible statement type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

10. **(Original)** The system of Claim 8, wherein the statement type value comprises a data value indicating that the statement is of a type selected from the group comprising: data, executable, and compiler.

11. **(Original)** The system of Claim 1, wherein the statement list is created by examining the relationship of a token in the token stream to other tokens preceding or following the token in question.

12. **(Original)** The system of Claim 1, wherein the software counting tool is further operable to:

parse a second file containing computer source code to create a second token stream in response to the one of the plurality of sets of configuration data,

wherein the second file comprises a different version of the source code contained in the first file,

create a second list of statements in response to the second token stream, and

compare the first list of statements to the second list of statements to generate at least one count responsive to differences between the first list of statements and the second list of statements.

13. **(Amended)** A method of counting lines of source code, comprising:

selecting one of a plurality of sets of configuration data,

wherein each set of the plurality of sets of configuration data is associated with at least one computer language,

wherein, collectively, the plurality of sets of configuration data are associated with a plurality of computer languages,

wherein the selected set of configuration data comprises the keywords for a first computer language,

parsing a first file containing computer source code written in the first computer language to create a first token stream in response to the selected set of configuration data,

creating a first list of statements in response to the first token stream, **at least some of the statements comprising a combination of two or more tokens**, and

generating a count value in response to the first list of statements.

14. **(Original)** The method of Claim 13, wherein the plurality of sets of configuration data includes a different set for each of the following computer programming languages: C++ and Cobol.

15. **(Original)** The method of Claim 13, wherein each token in the token stream comprises a string of one or more characters and wherein each token in the token stream is associated with a token type value.

16. **(Original)** The method of Claim 15, wherein the token type value comprises a data value indicating that the token is of a type selected from the group comprising: an operator, a comment, a constant, a keyword, and an identifier.

17. **(Original)** The method of Claim 15, wherein the possible token type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

18. **(Original)** The method of Claim 15, wherein the plurality of sets of configuration data includes a different set for each of at least five different programming languages.

19. **(Original)** The method of Claim 13, wherein each statement in the list of statements is further associated with a statement type value.

20. **(Original)** The method of Claim 19, wherein the possible statement type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

21. **(Original)** The method of Claim 17, wherein each statement in the list of statements is further associated with a statement type value and wherein the possible statement type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

22. **(Original)** The method of Claim 13, further comprising:
parsing a second file containing computer source code to create a second token stream
in response to the one of the plurality of sets of configuration data,
wherein the second file comprises a different version of the source code
contained in the first file,
creating a second list of statements in response to the second token stream, and
comparing the first list of statements to the second list of statements to generate at
least one count responsive to differences between the first list of statements and the second
list of statements.

23. **(Amended)** A source code line counting system, comprising:
a computer readable storage medium; and
a software counting tool stored on the computer readable storage medium comprising:
a plurality of configuration files, each associated with one or more of a
plurality of computer languages,
a tokenizer operable to parse a file containing computer source code written in
a first computer language to create a token stream,
wherein the ~~parser~~ tokenizer is operable to parse source code written
in any of the plurality of computer languages,
wherein the ~~parser~~ tokenizer creates the token stream in response to
the configuration file associated with the first computer language;
a first statement builder operable to create a list of statements in response to
the token stream, at least some of the statements comprising a combination of two or
more tokens, and
a counter operable to generate a count value in response to the list of
statements.

24. **(Original)** The system of Claim 23, further comprising:
a plurality of additional statement builders wherein each of the plurality of statement
builders are associated with one or more computer languages and operable to generate a
statement list in response to a token stream generated from a source code file written in the
associated one or more computer languages.

25. **(Original)** The system of Claim 23, wherein each token in the token stream comprises a string of one or more characters and wherein each token in the token stream is associated with a token type value.

26. **(Original)** The system of Claim 25, wherein the possible token type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

27. **(Original)** The system of Claim 25, wherein each statement in the list of statements is further associated with a statement type value.

28. **(Original)** The system of Claim 27, wherein the possible statement type values that can be chosen do not vary based upon the language that the computer source code in the first file was written in.

29. **(Amended)** A method of measuring changes in source code comprising:
parsing a first file containing computer source code to create a first token stream,
creating a first list of statements in response to the first token stream, **at least some of the statements in the first list of statements comprising a combination of two or more tokens,**

parsing a second file containing computer source code to create a second token stream,

creating a second list of statements in response to the second token stream, **at least some of the statements in the second list of statements comprising a combination of two or more tokens,** and

comparing the first list of statements to the second list of statements to generate at least one count responsive to a comparison between the first list of statements and the second list of statements.

30. **(Original)** The method of Claim 29, wherein the at least one count is equal or approximately equal to a total number of statements in the second list of statements that are modified versions of statements in the first list of statements.

31. **(Original)** The method of Claim 29, wherein the at least one count is equal or approximately equal to a total number of statements on the second list of statements that did not appear on the first list of statements.

32. **(Original)** The method of Claim 29, wherein the at least one count is equal or approximately equal to a total number of statements on the first list of statements that did not appear on the second list of statements.

33. **(Original)** The method of Claim 29, wherein the at least one count is equal or approximately equal to a total number of statements on the first list of statements that also appear on the second list of statements.

34. **(Original)** The method of Claim 29, wherein the second file comprises a modified version of the first file and wherein the comparing step generates data approximating or equal to each of the following:

- a total number of statements in the second list of statements that are modified versions of statements in the first list of statements,

- a total number of statements on the second list of statements that did not appear on the first list of statements,

- a total number of statements on the first list of statements that did not appear on the second list of statements, and

- a total number of statements on the first list of statements that also appear on the second list of statements.

35. **(Amended)** A source code line counting system, comprising:
a computer readable storage medium; and
a software counting tool stored on the computer readable storage medium and operable to
parse a first file containing computer source code to create a first token stream,
create a first list of statements in response to the first token stream, at least some of the statements in the first list of statements comprising a combination of two or more tokens,
parse a second file containing computer source code to create a second token stream,
create a second list of statements in response to the second token stream, at least some of the statements in the second list of statements comprising a combination of two or more tokens, and
compare the first list of statements to the second list of statements to generate at least one count responsive to a comparison between the first list of statements and the second list of statements.

36. **(Original)** The system of Claim 35, wherein the second file comprises a modified version of the first file and wherein the comparing step generates data approximating or equal to each of the following:
a total number of statements in the second list of statements that are modified versions of statements in the first list of statements,
a total number of statements on the second list of statements that did not appear on the first list of statements,
a total number of statements on the first list of statements that did not appear on the second list of statements, and
a total number of statements on the first list of statements that also appear on the second list of statements.

37. **(Amended)** A source code line counting system, comprising:
a computer readable storage medium;
a plurality of sets of configuration data, each set associated with at least one computer language, the plurality collectively associated with different computer languages; and
a software counting tool stored on the computer readable storage medium and operable to
receive a source code file, **and**
parse the source code file to create a token stream in response to one of the plurality of sets of configuration data,
create a list of statements in response to the token stream, at least some of the statements comprising a combination of two or more tokens, and
compute a statistical measure **in response to the list of statements, the statistical measure** related to the number of lines of source code in the source code file using the one of the plurality of sets of configuration data associated with the computer language that the source code in the source code file was written in.